

طراحی کامپایلر و زبان‌های

برنامه‌نویسی

تحلیل گرانحوی و لغوی

بهار ۱۴۰۳

دانشکده مهندسی برق و کامپیوتر

تمرین اول

1. به سوالات زیر پاسخ دهید.

الف) ۱- تفاوت اصلی میان statement های imperative و declarative را بیان کنید.

۲- آزمون زبان declarative را شرح دهید.

۳- توضیح دهید چطور زبان‌های برنامه‌نویسی pure-functional امکان موازی‌سازی بیشتری فراهم می‌کنند.

۴- توضیح دهید آیا همواره موازی‌سازی در زبان‌های pure-functional موجب بهینه‌سازی می‌شود.

ب) به دو روش imperative و declarative شبه کد تابعی را بنویسید که عدد n را ورودی گرفته و n امین عدد دنباله فیبوناچی را برگرداند.

پاسخ: الف) ۱- statement های imperative وضعیت برنامه (مقدار متغیرهای موجود) را تغییر می‌دهند در صورتی که declarative statement تنها مقادیر جدید تعریف می‌کنند.

- ۲- در یک scope با تعاریف مشخصی از x_1, x_2, \dots, x_n ، تمام موارد کاربرد یک عبارت مانند e که تنها شامل متغیرهای x_1, x_2, \dots, x_n باشد، باید مقداری یکسان داشته باشند.
- ۳- در زبان‌های برنامه‌نویسی pure-functional، می‌توانیم برای محاسبه یک تابع مانند f که عبارت‌های e_1, e_2, \dots, e_n را به عنوان ورودی دریافت می‌کند هر کدام از عبارت‌های e_1, e_2, \dots, e_n را به صورت موازی محاسبه کنیم زیرا مقادیر آن‌ها از یکدیگر مستقل هستند.
- ۴- خیر - برای مثال در یک switch case می‌توان عبارت موجود در هر case را به صورت موازی محاسبه کرد اما تنها یکی از این case ها اتفاق می‌افتد و محاسبه سایر case ها بیهوده است.

(ب)

```
def FibonacciImperative(n):  
    if n == 0:  
        return 0  
    elif n == 1:  
        return 1  
    a, b = 0, 1  
    for _ in range(2, n + 1):  
        a, b = b, a + b  
    return b
```

```
def FibonacciDeclarative(n):  
    if n == 0:  
        return 0  
    elif n == 1:  
        return 1  
    else:  
        return FibonacciDeclarative(n-1) +  
        FibonacciDeclarative(n-2)
```

2. با توجه به قوانین لغوی زیر بیان کنید که خروجی تحلیلگر لغوی برای ورودی "bbaacdabcaaabaa"

چیست؟ در صورت مواجهه با خطا، از روش panic mode استفاده کنید (علامت "?" به معنی وجود فقط 0 یا 1 تکرار از کاراکتر مورد نظر است).

$T_1: a?(b|c)*a$

$T_2: b?(a|c)*b$

$T_3: c?(b|a)*c$

پاسخ: چون d در عبارت‌های منظم این تحلیلگر لغوی وجود ندارد، با استفاده از روش panic mode آن را حذف می‌کنیم. خروجی باقی رسته به صورت زیر خواهد بود:

$T_3 T_1 T_2 T_1$

3. گرامر زیر را در نظر بگیرید:

$S \rightarrow AB \mid aaB$

$A \rightarrow a \mid Aa$

$B \rightarrow b$

آیا گرامر فوق مبهم است؟ در صورت مبهم بودن مثالی بزنید که برای آن دو اشتقاق وجود داشته باشد و درخت parse آن‌ها را رسم کرده و سپس گرامر را طوری تغییر دهید که ابهام از بین برود. در صورت مبهم نبودن گرامر توضیح دهید که چرا ابهام وجود ندارد.

پاسخ: برای رشته aab دو اشتقاق بصورت زیر وجود دارد:

$S \rightarrow AB \rightarrow AaB \rightarrow aaB \rightarrow aab$

$S \rightarrow aaB \rightarrow aab$

گرامر رفع ابهام شده بصورت زیر میباشد:

$S \rightarrow Ab$

$A \rightarrow a \mid Aa$

4. برای هر یک از موارد زیر، یک گرامر مستقل از متن ارائه دهید.

الف) رشته‌هایی با الفبای $\{a, b\}$ که تعداد a ها دقیقا دو برابر تعداد b ها می باشد.

مثال‌های مورد قبول:

ε

aba

abbaaa

مثال‌های غیر قابل قبول:

a

bba

ب) لیست های nested که الفبای آن تنها شامل $\{0, 1, [,], \}$ می باشد. دقت کنید که ویرگول جزو

الفبای زبان است. برخی از مثال‌هایی که در این گرامر پذیرفته میشوند بصورت زیر هستند:

[]

[1]

[[0], 1]

[[1, 0], 0, [[1]], []]

و برخی از مثال‌هایی که در این گرامر پذیرفته نمی شوند بصورت زیر می باشند:

ε

[0,

1

[11]

[1, 0,

پاسخ: الف)

$$S \rightarrow bSaSa \mid aSbSa \mid aSaSb \mid SS \mid \epsilon$$

(ب)

$$S \rightarrow [T]$$

$$T \rightarrow L \mid \epsilon$$

$$L \rightarrow P \mid P, L$$

$$P \rightarrow 0 \mid 1 \mid S$$

5. به سوالات زیر پاسخ دهید.

الف) یک DFA برای یک زبان برنامه‌نویسی طراحی کنید که هر id یا همان شناسه با یک حرف شروع شود و حداکثر دارای ۴ کاراکتر باشد. کاراکترهای دوم تا چهارم در صورت وجود عدد نیز می‌توانند باشند ولی تعداد رقم‌ها نباید از تعداد کاراکترها بیشتر باشد (مجموعه ارقام را D و مجموعه حروف را L در نظر بگیرید. مجموعه الفبای این زبان برنامه‌نویسی فقط ارقام و حروف است).

ب) فرض کنید در حال طراحی یک زبان برنامه‌سازی هستیم که برای نام‌گذاری متدهای عمومی (public methods) و خصوصی (private methods) آن از نمونه جملات زیر استفاده خواهیم کرد. دقت شود که متدهای خصوصی باید با _ شروع شوند و نباید با _ خاتمه یابند. در نام‌گذاری متدهای عمومی، نباید از _ در ابتدای نام متد استفاده کنیم. در این زبان برنامه‌نویسی، تنها حروف کوچک به همراه _ در نام‌گذاری‌ها استفاده می‌شوند و استفاده از حروف بزرگ و اعداد مجاز نیست. همچنین استفاده از دو عدد _ متوالی مجاز نمی‌باشد.

نمونه نام‌گذاری متدهای عمومی: add_all , compute , simple_method , simple_function_ , met_

نمونه نام‌گذاری متد خصوصی: _private_method , _pmethod

یک Expression Regular بنویسید که بتواند رشته‌های مجاز در نام‌گذاری متدهای عمومی و خصوصی را تولید نماید و سپس یک DFA بکشید که تمامی نام‌های مجاز را بپذیرد.

(ج) برای تمام رشته‌های باینری که شامل دو 1 هستند که فاصله‌ی آن‌ها تعداد فرد کاراکتر است، عبارت منظمی بنویسید و سپس NFA آن را ترسیم کرده و بعد آن را به DFA تبدیل کنید. (توجه کنید که حتماً مراحل بالا را طی کنید و هر بخش دارای نمره مختص به خودش می‌باشد). مثال:

001001100

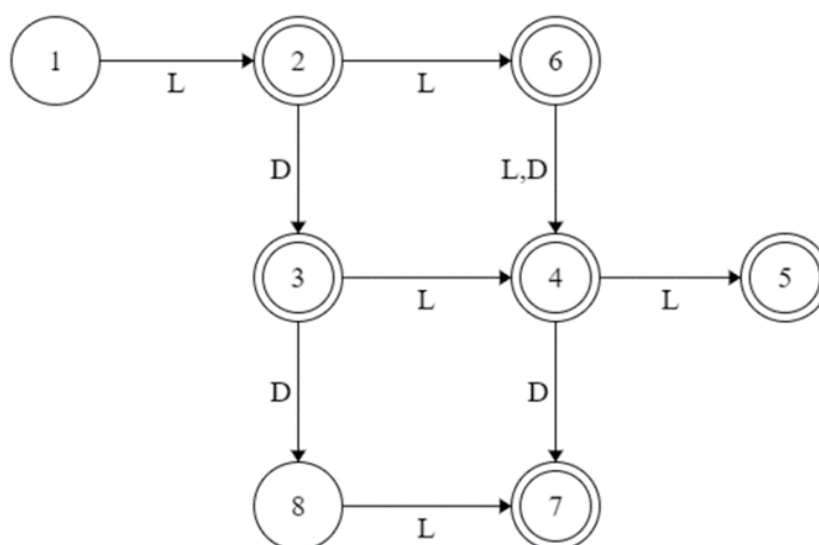
پاسخ: الف) با توجه به اینکه تعداد ارقام نباید بیشتر از حروف باشد و کاراکتر اول باید از حروف باشد، این id ها می‌توانند بدین شکل باشند:

4 کاراکتر: L L L L - L L L D - L L D L - L D L L - L D D L - L D L D - L L D D

3 کاراکتر: L L L - L L D - L D L

2 کاراکتر: L L - L D

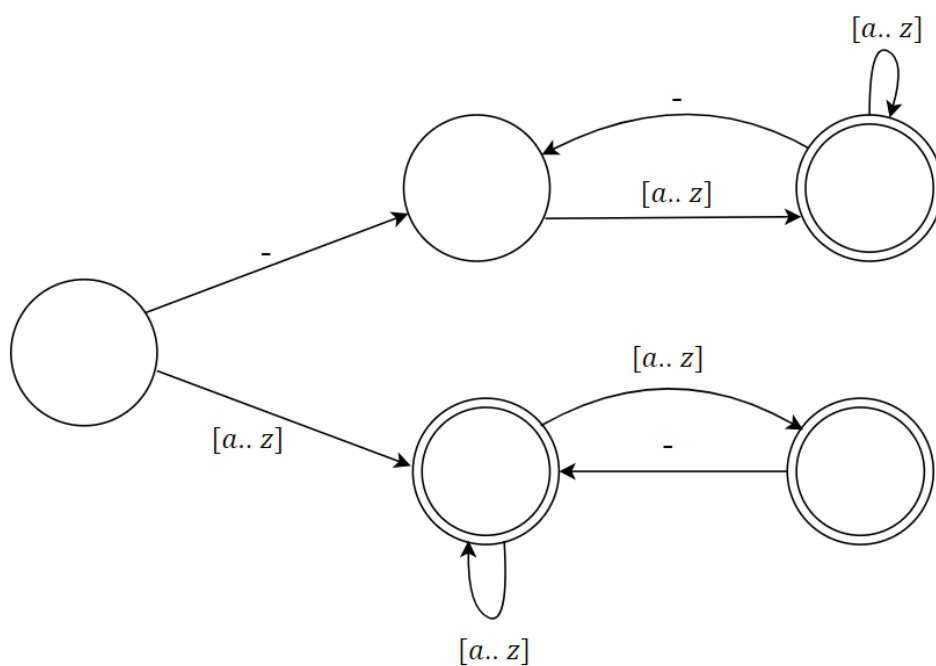
تک کاراکتر: L



ب) البته این سوال می‌تواند پاسخ‌های متنوعی داشته باشد. ولی یک پاسخ صحیح برای این سوال طبق راه حل زیر خواهد بود:

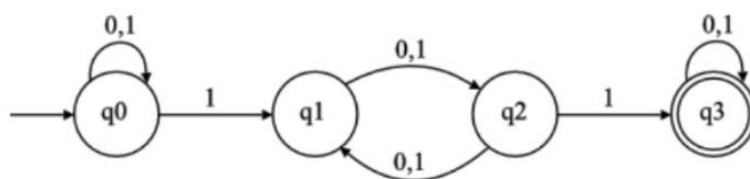
$$([a..z]^+)^+ \mid [a..z]^+ ([a..z]^+)^* (?)$$

$$([a..z]^+)^+ \mid ([a..z]^+ ([a..z]^+)^*)^+$$

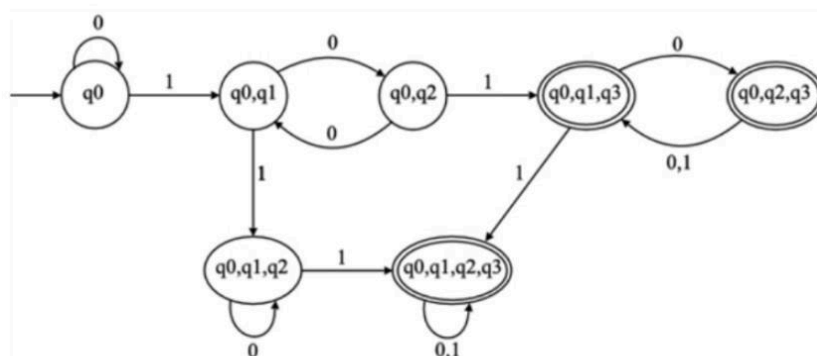


$$(1+0)^* 1(11+10+01+00)^* (1+0)1(1+0)^* \quad \text{ج)}$$

NFA:



DFA:



6. فرض کنید در حال طراحی یک زبان برنامه‌نویسی هستیم و می‌خواهیم در تحلیلگر لغوی دو token یکی برای مقادیر صحیح به نام INT_LITERAL و یکی برای مقادیر اعشاری به نام FLOAT_LITERAL تعریف کنیم. با توجه به قوانین موجود در زبان برنامه‌نویسی، که در پایین ذکر شده regular expression های مربوط به این دو token را بنویسید. (سعی کنید token های میانی ایجاد کنید تا در صورت خطا در regular expression های نهایی بخشی از نمره سوال را دریافت کنید).

قوانین مربوط به مقادیر عددی در زبان:

۱- اعداد صحیح می‌توانند در مبنای ۲ یا ۱۰ یا ۱۶ نوشته شوند که به ترتیب حروف 'b' و 'd' و 'h' در ابتدای آن‌ها قرار می‌گیرد.

۲- پس از حروف یک علامت '-' قرار می‌گیرد و سپس مقدار نوشته می‌شود. مثال: h-18

۳- در صورتی که یک مقدار بدون حرف قبل و '-' بیاید مبنای ۱۰ در نظر گرفته می‌شود. مثال: 18

۴- اعداد اعشاری از تمامی قوانین بالا پیروی می‌کنند با این تفاوت که برای بیان اعشاری بودن آن‌ها یا یک '.' در میان ارقام قرار می‌گیرد یا در انتهای عدد یک کاراکتر 'f' قرار داده می‌شود (دقت کنید که وجود '.' در میان ارقام و 'f' در انتهای عدد به طور هم‌زمان در یک عدد مجاز نیست).

مثال‌هایی از رشته‌های درست و نادرست با توجه به تعریف این token ها:

b-12	X	B4.12	X
h-12	✓	b-101.23	X
d98	X	b-101.100f	X
h-1A9F	✓	-12	X

پاسخ:

HEX : ['0'-'9','A'-'F']

DEC : ['0'-'9']

BIN : ('0'+'1')

HEX_VAL : 'h-'HEX⁺

DEC_VAL : ('d-' + ε)DEC⁺

BIN_VAL : 'b-'BIN⁺

HEX_FLOAT : HEX_VAL('.'HEX⁺ + 'f')

DEC_FLOAT : DEC_VAL('.'DEC⁺ + 'f')

BIN_FLOAT : BIN_VAL('.'BIN⁺ + 'f')

INT_LITERAL : (HEX_VAL + DEC_VAL + BIN_VAL)

FLOAT_LITERAL : (HEX_FLOAT + DEC_FLOAT + BIN_FLOAT)